2D Bayesian Fracture Videos: Generation and Inference

Anonymous ECCV submission

Paper ID ***

Abstract. Despite previous research in modeling physical properties from video evidence, no work for modeling the physical properties of rigid objects like pottery and logs as they undergo fracture exists to our knowledge. We note several key elements that should be included in any such physical model of fracture, namely, that the modeling is in 3-d, and that the following elements are included: (1) the fracture event, itself, (2) the geometry of the initial object, the number of fragments and their corresponding geometries, (3) the physical properties of the fragments, including position and momentum, and (4) collision between objects. To this end, we propose a probablistic graphical model that includes priors of the laws of physics and data in terms of pre-processed images (TODO what kind?). Inference over this model is done using (TODO inference technique). To evaluate this model and inference technique, we use the (TODO) method on the (TODO) dataset. Our initial experiments show that (TODO). . . .

Keywords: bayesian computer vision fracture tracking modeling physics

1 Introduction

Fracture events are of interest to many clients. Departments of transporation are responsible for the quick response to and recovery from quickly changing roadside conditions. For mountainous roads, this can include rock slides and the collapse of rock cliffs onto the roadway. Another class of organizations, mining companies, require prediction and modeling of the geology for possible failures as they extract minerals from the earth, changing the landscape in the process. Third, structural collapses such as dam failures can be catostrophic, and civil engineers would benefit from an ability to model and predict such collapses.

1.1 Key Model Features

Any insight provided to these clients is helpful, and, to do so, we must model the fracture event, including its underlying physical properties. While it is impossible to perfectly model physics with current computing capabilities, and it would be even more difficult to find a matching physical simulation given video evidence, we identify several key properties that a physical model for fracture should include and several simplifying assumptions to make the model more manageable.

First, the model must include the fracture event, itself. for a fracture to have occurred in the first place, a time span over which the fracture occurs is required. For rigid objects which shatter very quickly, as opposed to flexible objects that fail slowly over time (”ductile” failure), this time span becomes negligible in terms of the individual video frames. Thus, we make the simplifying assumption that the fracture occurs at a single point in time. In addition to ”time of fracture,” a fracture event includes fracture boundaries, which split the initial object into many fragments. Abstractly, this can be thought of as a partition of the object’s volume.

Second, the geometry of the objects should be included. For a fracture event, this means that an object with one geometry becomes many objects, based on the partition above. In theory, this partition could be arbitrary, but we make the simplifying assumption that partition can be determined by a recursive binary partition along boundaries that are roughly planar. Furthermore, we assume that no extremely small fragments, i.e., dust, exist. In reality, the existence of such small dust fragments are negligible, in comparison to the much larger fragments.

Third, the physical properties of the objects should be included. This includes positional properties, namely, the position and angle of the original object and each fragment’s center-of-mass and associated velocities, for each video frame. Here, we assume that the initial object is stationary. This simplifies the model in that, before the fracture, the only object that is moving is the object causing the fracture, whereas, in the general case, one or more of (1) the object which is fractured and (2) the object which is causing the fracture, may be moving. When the fracture happens, finding out the new positions and velocities of all the new fragments requires an additional physical property: the energy which was added by the object which caused the fracture. Aside from this added energy, we assume conservation of energy, and thus momentum, for the fragments. Further, assuming uniform density allows us to easily figure the energy added to each fragment using each fragment’s volume, in proxy of mass. After the fracture has happened, distance and velocity values should update deterministically from the physical laws of motion. In practice, though, this is unteneble. Therefore, we assume that probablistic updates occur at discrete time points. These probablistic updates can still incorporate physical priors (as the mode of the probablistic update).

Fourth, collision between objects should be modeled. In the general case, and especially with geometry that includes concavities, calculations for collision detection can be costly. If we begin with convex geometry or geometry that is decomposed into a partition of convex geometry, the assumption of planar fracture ensures that the only kind of geometry that we deal with is convex. This simplifying assumption makes collision detection much simpler, which also makes our machine learning approach more tenable.

1.2 Problem Statement

In the context of video evidence, the requirements of the physical fracture model outlined above require solving several difficult tasks. First, the time of fracture must be identified from the video sequence evidence. Initially, this may not seem like that difficult of a task, and, indeed, it is certainly not the most difficult problem. However, this can be complicated due to other movement in the scene, namely movement of the object causing the fracture.

Second, the number of fragments after the fracture event should be found, to model their geometry and motion accurately. This is made difficult due to the nature of fracture of rigid objects: Fractures typically occur quickly, between camera frames. In other words, the single object becomes many in the span of a frame. Occlusion also complicates this task, as some of the fragments could be occluded by others.

Another task is identifying the fragments' positions in the frame. The objects we are dealing with could be moving very fast and be very small. Even in simple video sequences with a stark difference in value between the object and the background, existing segmentation techniques may fail to segment the fragments from each other. Further complicating things, motion blur means that a single frame could provide evidence for a range of positions for each fragment. Even examining the frame sequences in their entirety, using motion of the image patches as evidence, traditional techniques for finding motion of image patches of sequences, like optical flow, have difficulty with such fast movement of such small objects.

In addition, physical modeling requires a notion of the geometry of the object. Typical surface reconstruction techniques, like structure from motion, require multiple views to recover a point cloud that represents the geometry of the scene, and, even then, further processing is required to obtain an object volume.

Last, in addition to geometry, properties like position and momentum in 3-d require a notion of pysics, to connect the evidence from the previous tasks to the physical properties. However, deterministic physics simulation suffers from chaos theory. Small changes in the initial conditions produce wildly different outputs.

1.3 Contributions

We make several contributions toward modeling fracture events from video evidence. First, we identify key features that a fracture model should have and the potential difficulties in implementing such a model. Second, we devise a probablistic graphical model that handles some of the difficulties above. Finally, we identify an adequate inference technique for the defined model.

2 Related Work

2.1 Tracking

While we frame our proposal in terms of physical modeling, there is some overlap between our goal and tracking. Tracking is a broad field of research, and many techniques exist which produce a variety of object representations in the image plane. The most prominent object representation in use is bounding boxes, but Yilmaz et al.[?] note two object representations that are closer to the output of our model: primitive geometric shapes, as used in [?] and object silhouette, as used in [?].

Statistical methods have been employed for tracking in [?] [?] [?] [?]. Brau et al.[?] use a generative, graphical model which models people as cylinders as they move within the frame and the images from the perpective of a simplified, three parameter camera.

Deep learning has been leveraged extensively for tracking problems.[?] Deep learning approaches generally rely extensively on labeled data. This means that the data sets available and the labels provided affect the kinds of outputs that such deep learning approaches can produce. Importantly, the vast majority of deep learning approaches produce bounding box representations of the objects. Instead, we seek to represent the objects as 3-d geometry.

Overall, we found no existing tracking techniques designed to handle a single object splitting into multiple objects.

2.2 Physical Modeling

Some work has been done in using video evidence to infer various physical properties of objects, from position, to velocity, to mass, to more complicated properties. In “Physics 101: Learning Physical Object Properties from Unlabeled Videos,” 2016, Wu et al. used a convolutional neural network in conjunction with hard-coded physics equations to infer a myriad of physical properties, from mass to coefficients of friction, in a variety of scenarios, from ramps to springs to a liquid scenario.

In “What Players do with the Ball: A Physically Constrained Interaction Modeling,” 2016, Maksai et al. use probabilistic graphical modeling to introduce physical constraints into the task of tracking in sports videos, made difficult due to the balls’ small size and quick speed.

Many other papers exist in this domain, but, in our literature review thus far, we have seen none tackle the task of inferring physical properties in a fracture scenario.

3 Technique

Our strategy for tackling this problem has been to start with simpler versions of the problem and work our way up from there. It is important to identify three different dimensionalities that can each be moved up or down to complicate or simplify the model: (1) the dimensionality of the world, (2) the dimensionality of the object being fractured, and (3) the dimensionality of the space onto which the world will be projected.

For fracture of arbitrary, real objects in the real world, both (1) and (2) would be 3-d. However, it is also important to note that certain real-world objects like sheets of glass can be represented in lower dimensions. Our simplifications have focused on lower dimesionality along (1) and (2), while leaving (3) 2-d (images). So far, we have only worked in 2-d for (1). However, we have varied the dimensionality of the object.

We began with a 0-d object, which can be thought of as a single particle. This allowed us to perform position and velocity updates according to initial position and velocity random variables and a fixed gravitational constant. From there, we moved onto a 1-d object, a path in 2-d space, with the restriction that the path was straight. The existing physics updates for the 0-d model were incorporated as the stick's center of mass. Additionally, we added angle and angular velocity (and corresponding random variables) to the model. From here, we added fracture to the model, beginning with a single fracture. At this point, we allowed the fracture to occur at an arbitrary frame of the video, which meant that the model had changing dimensionality. We shortly added a binary fracture model, in which a single stick splits into at most two sticks in a single frame of the video and that the stick ancestry could be represented by a full binary tree. The corresponding Bayes net can be seen in the nearby figure. A model (with 2-d blocks) that is very similar to the model in the figure will later be explained.

From here, some preliminary work was done to explore potential inference techniques that we could use. We devised an experiment using Metropolis-Hastings random walk to infer the continuous variables while leaving the discrete variable, the time of fracture, fixed at its known value. In order to ensure that the inference worked at all, we devised a noise model that ensured that the observations for any data set had support over the space of priors. To do so, we added IID noise to each observed end point in the image space. Unfortunately, our results using Metropolis-Hastings were initially poor. We ran inference with chain lengths of 100,000. We had to use narrow standard deviations for the proposal distributions to achieve a reasonable acceptance rate, but this also meant that the chains were slow to converge and were prone to getting caught in local optima.

3.1 2-d Block, 2-d World Model

At this time, we moved onto a model with 2-d objects to better approximate the data sets that we were interested in at the time, firewood cutting videos. We still maintained a 2-d world. While this model may seem to complicate things in comparison to the 1-d stick model, several assumptions were made to simplify the 2-d objects. We enforced that the objects were rectangles (blocks), oriented orthogonally the world's standard basis. We also took a step back and simplified the fracture event. We enforced that the block did not move until the fracture happened. This allowed us to collapse all frames before the fracture to a single frame. In other words, in this model, there is no change of dimensionality. We also incorporated the conservation of momentum and angular momentum into this model, enforcing that, since the object is initially stationary, the sum of the momenta from the two fragments is 0.

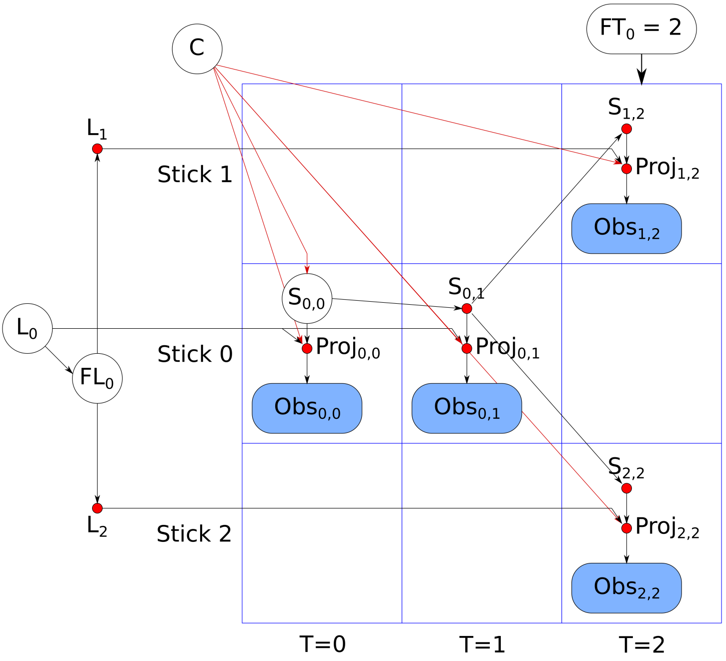

Fig. 1. The Bayesian Network for the 2-d world, 1-d object (stick) model. Here, there are three video frames and three objects: one parent object and two fragments. The fracture time has already sampled from a discrete distribution: $FT_0 = 2$. A very similar model, the 2-d block model, is explained later.

We explain this model in detail. A large portion of the network is dedicated to physical modeling, but the model must also incorporate the video sequence. This includes a camera as well as the pre-processed data from images that that camera produced. First, we define several values that are given as hyper-parameters:

- I_w: width of the images
- I_h: height of the images
- M: number of images
- C_{fps}: frames per second of the camera

and the calculated image aspect ratio and gravity constant:

$$I_{ar} = \frac{I_w}{I_h}$$

$$a_y = -\frac{9.8}{(C_{fps})^2}$$

Next, we sample a camera. For a 2-d embedding space, this is as simple as sampling the top of the camera, in world units – meters – and calculating the right boundary from that and the image aspect ratio:

$$\begin{aligned} C_l =& 0 \\ C_b =& 0 \\ C_t \sim& TruncNormal(\mu = 3, \sigma = 1, low = 0, high = \infty) \\ C_r =& C_t \times I_{ar} \end{aligned}$$

From these values, we calculate the meters per pixel and define a linear transformation from world coordinates to image pixels:

$$\begin{aligned} \frac{m}{pix} =& \frac{C_t - C_b}{I_h} \\ C_{mat} =& \begin{bmatrix} 0 & -\frac{pix}{m} & \frac{C_t \times pix}{m} \\ \frac{pix}{m} & 0 & -\frac{C_l \times pix}{m} \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

We can also sample the block dimensions and positions in terms of world coordinates, meters, using priors on the shapes and sizes of logs that are used for firewood. We use the notation $S_{n,m}$, a record over fragment ID N and timestamp M, each indexed from 0, which represents the state of that fragment's center of mass at that timestamp. Each such record contains both position, q, velocity, v, angle, Θ, and angular velocity, ω of which non-angular values are indexed by dimension, (x, y). For example, $S_{3,4}.q_y$ indicates the y position for the third fragment at the fourth frame. L_n is a record representing the geometry for block n, including w, h as well as a g, calculated geometry, a sequence of points in 2-d homogeneous coordinates for the block, in a space relative to the block's center of mass, and v, the volume of the block.

$$\begin{aligned}
S_{0,0}.q_x \sim & Normal(\mu = \frac{C_l + C_r}{2}, \sigma = \frac{C_r - C_l}{16}) \\
S_{0,0}.q_y \sim & Normal(\mu = \frac{C_t + C_b}{2}, \sigma = \frac{C_t - C_b}{16}) \\
S_{0,0}.v = & \overrightarrow{0} \\
S_{0,0}.\Theta = & 0 \\
S_{0,0}.\omega = & 0 \\
L_0.w \sim & TruncNormal(\mu = 0.2, \sigma = 0.075, \\
& low = 0, high = \infty) \\
L_0.h \sim & TruncNormal(\mu = 0.3, \sigma = 0.2, \\
& low = 0, high = \infty) \\
L_n.g = & \begin{bmatrix} \frac{-L_n.w}{2} & \frac{L_n.w}{2} & \frac{L_n.w}{2} & \frac{-L_n.w}{2} \\ \frac{L_n.h}{2} & \frac{L_n.h}{2} & \frac{-L_n.h}{2} & \frac{-L_n.h}{2} \\ 1 & 1 & 1 & 1 \end{bmatrix} \\
L_n.v = & L_n.w \times L_n.h
\end{aligned}$$

We move onto the fracture event, itself. Since we assume the initial block is stationary, we model the fracture event as always happening at $m = 1$. Currently, we do not model any additional energy added to the system. However, we do sample a momentum and angular momentum for the left block (block 1), and, using conservation of momentum, determine the corresponding momenta of the right block (block 2). For objects on a platform, we assume that there is no y momentum added. We also sample the fracture location for the vertically-oriented fracture plane:

$$\begin{aligned}
mom_{1,x} \sim & TruncNormal(\mu = 0, \sigma = 0.05, \\
& low = -\infty, high = 0) \\
mom_{2,x} = & -mom_{1,x} \\
angmom_1 \sim & TruncNormal(\mu = 0, \sigma = 0.2, \\
& low = 0, high = \infty) \\
angmom_2 = & -angmom_1 \\
FL_0 \sim & TruncNormal(\mu = \frac{L_0.w}{2}, \sigma = \frac{L_0.w}{4}, \\
& low = 0, high = L_0.w) \\
L_1.w = & FL_0 \\
L_2.w = & L_0.w - FL_0 \\
L_1.h = & L_2.h = L_0.h
\end{aligned}$$

After sampling the fracture variables, we calculate the initial states of the new blocks' centers of mass and the new blocks' geometry. This is straightforward.

We use the volume (area in 2-d) of the blocks and the momenta to calculate the added velocities. From here, we use Euler's method to determinstically update each block's state variables:

$$
\begin{aligned}
S_{n,m}.q_x =& S_{n,(m-1)}.q_x + S_{n,(m-1)}.v_x \\
S_{n,m}.q_y =& S_{n,(m-1)}.q_y + S_{n,(m-1)}.v_y \\
S_{n,m}.v_x =& S_{n,(m-1)}.v_x \\
S_{n,m}.v_y =& S_{n,(m-1)}.v_y + a_y \\
S_{n,m}.\mathit{\Theta} =& S_{n,(m-1)}.\mathit{\Theta} + S_{n,(m-1)}.\omega \\
S_{n,m}.\omega =& S_{n,(m-1)}.\omega
\end{aligned}
$$

From here, we project the points of each block into world coordinates via a transformation in terms of the position and angle of that block at that frame. Then, we project those coordinates into image coordinates using the camera matrix above, call these points $Act_{n,m,o,p}$, where n, m are indexed as above and o, p are point index and dimension index, respectively. Finally, we add some IID noise to those points:

$$
Obs_{n,m,o,p} \sim Normal(\mu = Act_{n,m,o,p}, \sigma = \frac{\sqrt{I_w \times I_h}}{512})
$$

We are currently exploring Metropolis-Hastings over this model. While experiments have been run, I need to do some data wrangling to reach any meaningful conclusions. After using multiple chains and increasing the chain length to 1,000,000, some of the results look promising in isolation, but it remains to be seen whether it is an adequate technique for this model in general.

4 Evaluation

5 Conclusion

Page 10 of the manuscript.

Page 11 of the manuscript.

Page 12 of the manuscript.

Page 13 of the manuscript.

Page 14 of the manuscript.

Page 15 of the manuscript.

Page 16 of the manuscript. This is the last page of the manuscript.

Now we have reached the maximum size of the ECCV 2018 submission (excluding references). References should start immediately after the main text, but can continue on p.15 if needed.